Re-Code (C++)

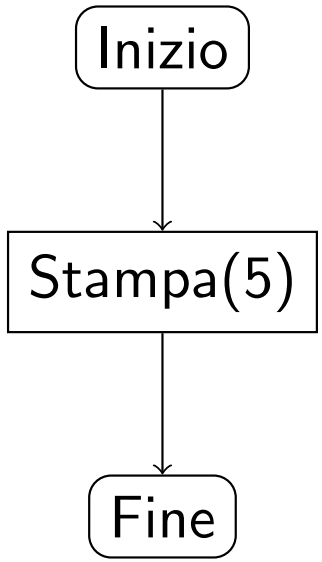
Salvatore Capolupo

April 17, 2025

Cosa sono i Diagrammi di Flusso

- Un diagramma di flusso una rappresentazione grafica di un algoritmo, passo dopo passo.
- Usa simboli standard per mostrare i passi da eseguire.
- Permette di visualizzare in modo chiaro la logica di un software o di procedimento.

Stampare un Numero



- In ogni algoritmo c' `sempre` un inizio e una fine. Non bisogna essere vaghi o scrivere le cose "tanto per".
- Stampa(5) denota una `logica "funzionale"`: `stampa()` una `funzione` astratta che prende come parametro un numero - il 5 - e lo stampa a video.

Non so usare C++ ... :-(

- ... devi stare calmo :-)
- `#include <iostream>` la **direttiva** che include una libreria per gestire input/output. Ci permetter di usare cin e cout.
- Un **preprocessore** serve invece a elaborare la direttiva prima che il codice venga compilato ed eseguito.

Scrivere "Hello, world!"

```
1 #include <iostream> // direttiva I/O
2 using namespace std; //evita di farci scrivere
      std::cout al posto del semplice cout
3
4 int main() { // il main restituisce un int(ero), che
      rappresenta l'esito del programma: esempio 0 OK, 1
      errore
5      cout << "Ciao, Mondo Dolce e Amaro!" << endl;
6      return 0;
7 }
```

```
1 << serve a concatenare gli output
2 endl significa "fine riga", ovvero end-l(ine)
```

Cosa facciamo adesso?

1. Stampare messaggi (Hello world)
2. Usare variabili e input da tastiera.
3. Calcolare operazioni matematiche (somma, media, ...).
4. Capire i cicli (for, while).
5. Comprendere le condizioni (if/else).
6. Risolvere un problema pi complesso
7. Risolvere il FizzBuzz.

Usare variabili e input da tastiera.

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      // Prima dichiaro...
5      int numero;
6      string nome;
7
8      cout << "Come ti chiami? ";
9      // ...poi uso
10     cin >> nome;
11
12     cout << "Dammi un numero: ";
13     // ...poi uso
14     cin >> numero;
15
16     // Output dei valori inseriti
17     cout << "Ciao " << nome << "! Hai inserito il
           numero " << numero << "." << endl;
18
```

Calcolare operazioni matematiche (somma, media, ...).

```cpp
#include <iostream>
using namespace std;
int main() // anche qui: prima dichiaro...
    int a, b, c;
    cout << "Inserisci tre interi: ";
    // ...poi uso
    cin >> a >> b >> c;
    int mediaIntera = (a + b + c) / 3; //
    cout << "La media intera e': " << mediaIntera <<
        endl;

    double x, y, z;
    cout << "Ora invece inserisci tre decimali: ";
    cin >> x >> y >> z;
    double mediaDecimale = (x + y + z) / 3;
    cout << "La media decimale e': " << mediaDecimale
        << endl;
    return 0;
}
```

Dichiarazione e uso variabili

```
1  #include <iostream>
2  using namespace std;
3
4  // delimitato da { e }
5  int main(){
6      // sbagliato
7      x = x+1; // ??
8      int x;    // andava prima... :-(
9
10     // sbagliato
11     int x; // inizializzato, non valorizzato
12     x= x+1; //??
13
14     //corretto
15     int x=10; //inizializzato, valorizzato
16     x= x+1; //adesso x conterr  11
17 }
```

Cosa e' un Blocco di codice

```
1  #include <iostream>
2  using namespace std;
3
4  // un blocco viene delimitato da { e }
5  int main(){
6    // riga 1
7    // riga 2
8    // ...
9    // riga n
10 }
```

Ciclo for

```
#include <iostream>
using namespace std;

int main(){
for ( int indice = da; indice <= fino_a; indice++ )
{
  // riga 1
  // riga 2
  // ...
  // riga n
}
}
```

Ciclo while

```
#include <iostream>
using namespace std;

// delimitato da { e }
int main(){
int indice = 0;
while (indice <= 10){
 // riga 1
 // riga 2
 // ...
 // riga n
indice++; //aumenta a passo 1, come nel for
}
}
```

if / else

```
#include <iostream>
using namespace std;

int main(){
 int a = 5, b= 6;
 if ( a%2==0 )
    cout<<"Numero pari";
 else
    cout<<"Numero dispari";
}
```

if innestati

```
#include <iostream>
using namespace std;

int main(){
if (eta>=18){
if (matricola)
    segui_corsi(1); //1     l'anno accademico
else
    segui_corsi(2);
}
}
```

swap (scambio)

```
1  #include <iostream>
2  using namespace std;
3
4  // delimitato da { e }
5  int main(){
6  float x = 4.0, y= 10.0, tmp = 0;
7  x = y;
8  y = x; //sbagliato
9
10 tmp = x;
11 x = y;
12 y = tmp; //ho scambiato 4.0 e 10.0
13 }
```

Cicli in C++

- I cicli permettono di eseguire un blocco di codice piu di una volta.
- Posso farli con **for** oppure **while**.

Se uso il for

```
for (int i = 0; i < 5; i++) {
cout << "Iterazione " << i << endl;
}
```

Se uso il while

```
int i = 0;
while (i < 5) {
cout << "Iterazione " << i << endl;
i++;
}
```

Cicli (loop)

- **for**: il numero di iterazioni si conosce a priori.
- **while**: il numero di iterazioni dipende / non si conosce.
- Nota: **break** per terminare - **continue** per saltare iterazioni.

Sommare i primi 5 numeri con for

```
1 int somma = 0;
2 for (int i = 1; i <= 5; i++) {
3 somma = somma + i; // 1 + 2 + 3 + 4 + 5
4 }
```

Da inserire entrambi dentro un main(), ma ormai sappiamo come si fa. Giusto :-) ? **Sommare i primi 5 numeri con while**

```
1 int somma = 0, i = 1;
2 while (i <= 5) {
3 somma = somma + i; // 1 + 2 + 3 + 4 + 5
4 i++;
5 }
6 //somma viene detta variabile accumulatore
```

Swap tra due variabili

Scambio errato senza variabile temporanea:

```
int x = 5, y = 3;
x = y;
y = x;
// Errore: x e y avranno lo stesso valore
```

Scambio corretto con variabile temporanea:

```
int x = 5, y = 3, tmp;
tmp = x;
x = y;
y = tmp;
```

Scambio errato (senza variabile temporanea)

Codice:

```
1     int x = 5, y = 3;
2     x = y;
3     y = x; // Errore: x e y coincidono
```

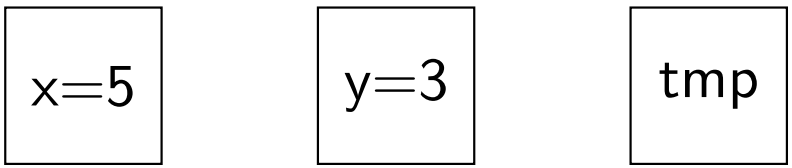
x=5
x=3

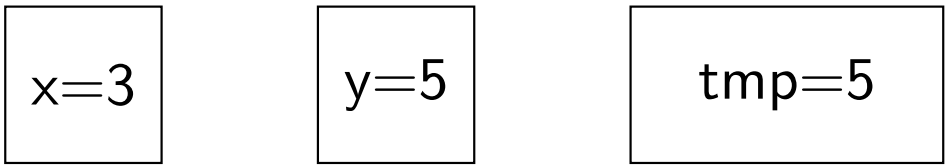
y=3
y=3

Scambio corretto (con variabile temporanea o di backup)

Codice:

```
1     int x = 5, y = 3, tmp;
2     tmp = x;
3     x = y;
4     y = tmp;
```





Un problema (un po') piu difficile

1. Mario tifa una squadra di serie A che gioca, risaputamente, un totale di 19 partite in casa.
2. Vorrebbe comprare almeno un biglietto per partita, da 1 a massimo 5 alla volta.
3. Obiettivo: calcolare quanto ha speso complessivamente per tutti i biglietti.
4. Risoluzione: il biglietto di ciascuna partita ha un prezzo unitario che deve essere moltiplicato per il numero di biglietti acquistati per quella specifica partita, usando una variabile accumulatore inizialmente a zero.

Un problema (un po') piu difficile

1. Diagramma di flusso?

Proposta di soluzione

```cpp
#include <iostream>
using namespace std;
int main() {
    const int P = 19;
    const double PREZZO_BIGLIETTO = 50.0;
    double totale = 0.0;
    for (int i = 1; i <= P; i++) {
        int biglietti;
        cout << "Partita " << i << ": Quanti
            biglietti desideri? (1-5, digita 0 per
            uscire) ";
        cin >> biglietti;

        totale = totale + biglietti *
            PREZZO_BIGLIETTO;
    }
    cout << "Mario ha speso: " << totale;
    return 0;}
```

Fizz-Buzz

```
1
2 Considera gli interi da 1 a 100 e:
3     Stampa "Fizz" per i numeri divisibili per 3.
4     Stampa "Buzz" per i numeri divisibili per 5.
5     Stampa "FizzBuzz" per i numeri divisibili sia per
          3 che per 5.
6     Solo nel caso in cui il numero non rientri in
          nessuno dei casi precedenti, stampalo.
7     I requisiti del problema vanno rispettati
          tassativamente.
```

Fizz-Buzz

```
Output desiderato:

1
2
Fizz
4
Buzz
Fizz
7
8
...
...
97
98
Fizz
Buzz
```

Fizz-Buzz

```
1 Problemi classici nella codifica del fizz-buzz:
2
3 1) necessario seguire un ordine nelle if: devi prima
     verificare la condivisione di divisibilita' per 3
     e per 5, poi le altre
4 2) l'uso dell'else va pesato: se lo usi senza che sia
     necessario, la soluzione sara' errata
5 3) necessario rispettare tutto quello che abbiamo
     visto finora
6 4) possibile implementare la soluzione in molti modi
     diversi, tutti equivalenti
7 5) le migliori implementazioni usano poche righe di
     codice (nel caso ottimo, giusto un paio)
```

Fizz-Buzz (proposta)

```cpp
#include <iostream>
using namespace std;
int main() {
    for (int i = 1; i <= 100; i++) {
        if (i % 3 == 0 && i % 5 == 0) {
            cout << "FizzBuzz" << endl;
        } else if (i % 3 == 0) {
            cout << "Fizz" << endl;
        } else if (i % 5 == 0) {
            cout << "Buzz" << endl;
        } else {
            cout << i << endl;
        }
    }
    return 0;
}
```

Fizz-Buzz (piu breve)

```
#include <iostream>
using namespace std;

int main() {
    for (int i = 1; i <= 100; i++)
        cout << (i % 3 == 0 ? "Fizz" : "") << (i % 5
            == 0 ? "Buzz" : "") << (i % 3 && i % 5 ?
            to_string(i) : "") << endl;
    return 0;
}
```